



(19)

(11) Publication number: **06348**

Generated Document.

PATENT ABSTRACTS OF JAPAN(21) Application number: **05167461**(51) Intl. Cl.: **G06F 7/50 G06F 7/52**(22) Application date: **14.06.93**

(30) Priority: (43) Date of application publication: 22.12.94 (84) Designated contracting states:	(71) Applicant: NIPPON TELEGR & TELEPH <NTT> (72) Inventor: MINAMI TOSHIHIRO KASAI RYOTA (74) Representative:
--	---

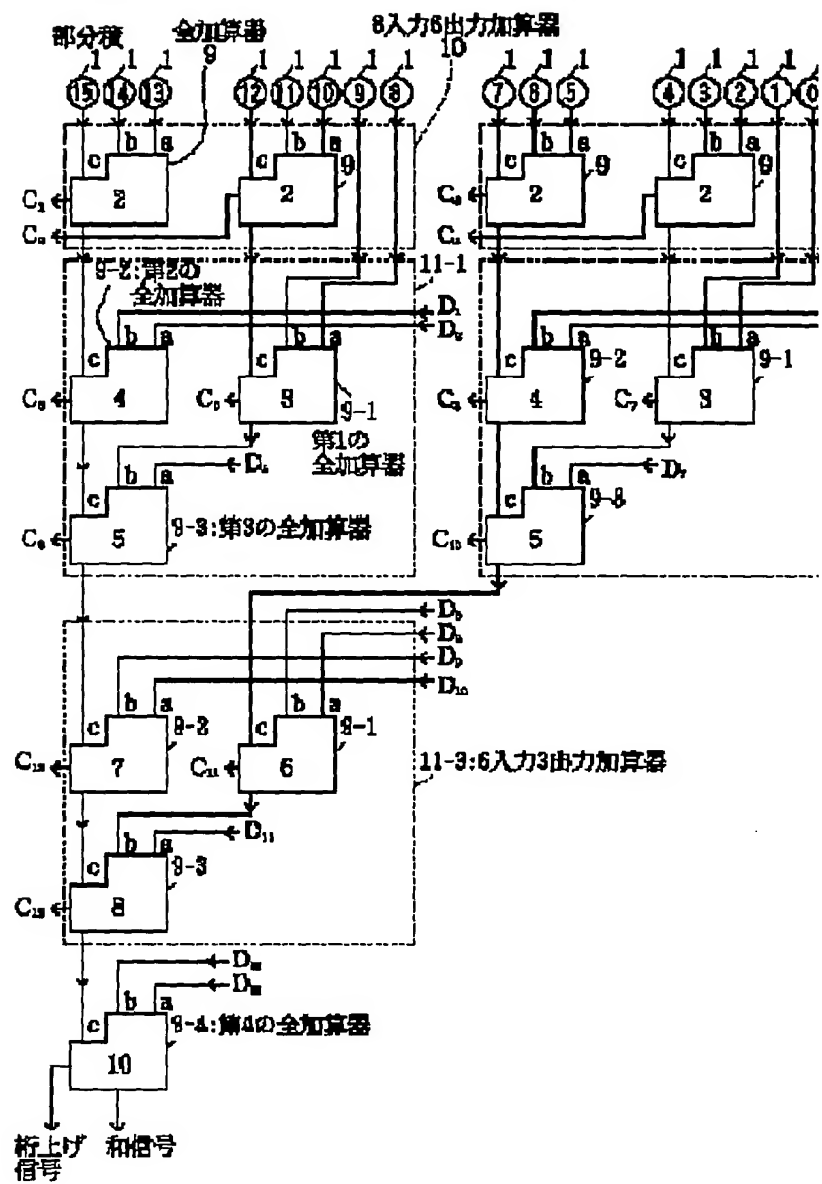
**(54) WALLANCE TREE
TYPE CARRY HOLDING
ADDER**

(57) Abstract:

PURPOSE: To provide a Wallace tree type carry holding adder capable of reducing the stage number of exclusive OR gates in a critical path without increasing the quantity of hardware as compared with a case of using a full adder as a constitutional unit and obtaining regular wiring similar to a case of using a 4-input 2-output adder as a constitutional unit.

CONSTITUTION: Since a full adder 9 having plural input terminals (a), (b) and (c) has a property that the delay time of a path for outputting a sum signal or a carry signal inputted from an input terminal (c) is a half of the delay time of a path for outputting a sum signal or a carry signal inputted from an input terminal (a) or (b), the property is utilized for constituting a 6-input 3-output adder 11 by using three full adders 9 and the Wallace tree type carry holding adder is constituted by using the adder 11 as a constitutional unit.

COPYRIGHT: (C)1994,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平6-348457

(43) 公開日 平成6年(1994)12月22日

(51) Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 7/50	G			
	N			
7/52	3 1 0 F			

審査請求 未請求 請求項の数 1 F D (全 12 頁)

(21) 出願番号 特願平5-167461

(22) 出願日 平成5年(1993)6月14日

(71) 出願人 000004226

日本電信電話株式会社

東京都千代田区内幸町一丁目1番6号

(72) 発明者 南 俊宏

東京都千代田区内幸町1丁目1番6号 日

本電信電話株式会社内

(72) 発明者 笠井 良太

東京都千代田区内幸町1丁目1番6号 日

本電信電話株式会社内

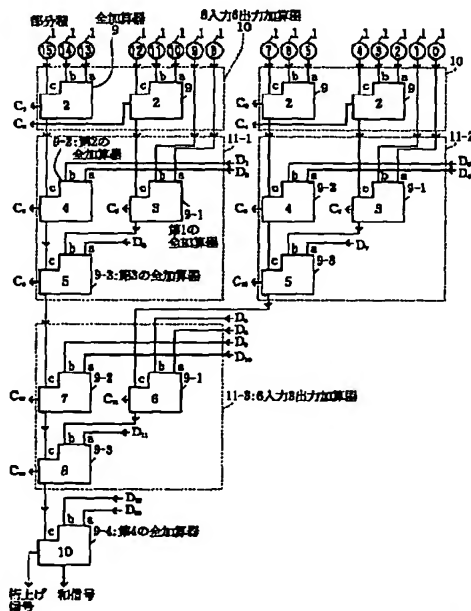
(74) 代理人 弁理士 川久保 新一

(54) 【発明の名称】 Wallace トリー型桁上げ保存加算器

(57) 【要約】

【目的】 全加算器を構成単位とした場合に比べてハードウェア量の増加がなく、クリティカルパスにおける排他的論理和ゲートの段数を削減することができ、また、4入力2出力加算器を構成単位とした場合と同様に規則的に配線することができるWallace トリー型桁上げ保存加算器を提供することを目的とするものである。

【構成】 全加算器が入力端子 a、b、c を有し、入力端子 c から入力して和信号または桁上げ信号を出力するパスの遅延時間が、入力端子 a または入力端子 b から入力して和信号または桁上げ信号を出力するパスの遅延時間の半分である性質があり、3個の全加算器を使用して6入力3出力加算器を構成するときに上記性質を利用し、6入力3出力加算器を構成単位としてWallace トリー型桁上げ保存加算器を構成したものである。



23707

【特許請求の範囲】

【請求項1】 m桁の被乗数とn桁の乗数とを乗算する場合、またはm桁の2進数をn個加算する場合に用いられ、nビットの入力信号を加算する過程で中間桁上げ信号が発生すると1桁上位のWallace トリー型桁上げ保存加算器に送り、1桁下位のWallace トリー型桁上げ保存加算器において中間桁上げ信号が発生するとそれも加算して最後に1ビットの和信号と1ビットの桁上げ信号を出力するWallace トリー型桁上げ保存加算器において、
 上記nビットの入力信号を8ビットごとに分ける手段と；全加算器2個、または全加算器2個と半加算器1個とによって8入力6出力加算器を構成し、全加算器2個で上記8入力6出力加算器を構成する場合には、同一桁で加算される4ビットの信号と1桁上位で加算される2ビットの桁上げ信号とを出力し、一方、全加算器2個と半加算器1個とで上記8入力6出力加算器を構成する場合には、同一桁で加算される3ビットの信号と1桁上位で加算される3ビットの桁上げ信号とを出力する手段と；6入力3出力加算器が第1の全加算器と第2の全加算器と第3の全加算器とで構成され、上記6入力3出力加算器へ入力する6ビットの信号を、上記第1の全加算器と上記第2の全加算器とに送り、上記第1の全加算器の和信号を、上記第3の全加算器の3個の入力端子のうち遅延時間の長い2個の端子の一方に送り、上記第1の全加算器の桁上げ信号を、1桁上位のWallace トリー型桁上げ保存加算器内の上記6入力3出力加算器に送り、上記第3の全加算器の3個の入力端子のうち遅延時間の長い2個の端子の他方には、1桁下位のWallace トリー型桁上げ保存加算器内の6入力3出力加算器から送られてくる桁上げ信号を入力し、上記第2の全加算器の和信号または1桁下位のWallace トリー型桁上げ保存加算器内の6入力3出力加算器における第2の全加算器から送られてくる桁上げ信号を上記第3の全加算器における遅延時間の短い端子に入力する手段と；上記8入力6出力加算器を最上段に置き、この8入力6出力加算器の下に上記6入力3出力加算器を置き、この6入力3出力加算器2個から出力される6ビットの信号を下段の6入力3出力加算器に入力することを繰り返して上記入力信号を3ビットにまとめる手段と；最後に上記3ビットの信号を入力して和信号と桁上げ信号とを出力する第4の全加算器と；上記6入力3出力加算器に入力する6ビットの信号のうち、上段の上記8入力6出力加算器または6入力3出力加算器中の遅延時間の短い経路を通過する2ビットの信号を、上記6入力3出力加算器中の第1の全加算器の3個の入力端子のうち遅延時間の長い2個の端子に入力する手段と；を有することを特徴とするWallace トリー型桁上げ保存加算器。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、乗算器や複数の入力を

一度に加算する複数オペランド加算器で用いられるWallace トリー型桁上げ保存加算器に関するものである。

【0002】

【従来の技術】 図7は、第1の従来例であるWallace トリー型桁上げ保存加算器を示すブロック図である。

【0003】 乗算器では、乗数と被乗数とによって部分積を生成し、この部分積を加算して積を求める。この部分積加算においては、まず、桁上げ保存加算器によって、複数の部分積が加算され、和信号と桁上げ信号とが生成され、次に、2入力加算器によって、それらの和信号と桁上げ信号とが加算される。

【0004】 図7には、16個の部分積1を加算する場合の例を示してあるが、各桁ごとに全加算器2をトリー状に配置して各桁内の加算を並列に行う。C1～C13は、1桁上位への中間桁上げ出力信号、D1～D13は、1桁下位からの中間桁上げ入力信号である。つまり、中間桁上げ出力信号C1～C13は、1桁上位の中間桁上げ入力信号D1～D13になる。

【0005】 図8は、上記従来例における全加算器2の一例である全加算器2-1を示す回路図であり、図9は、全加算器2の他の例である全加算器2-2を示す回路図である。

【0006】 全加算器2-1または2-2において、排他的論理和ゲート3における遅延時間は、通常、NANDゲートの遅延時間の2倍よりも長く、また、セレクトの遅延時間と同程度である。そこで、以下の説明では、最悪遅延時間を決めるクリティカルパスを、排他的論理和ゲート3の段数で評価することにする。

【0007】 この評価基準によると、全加算器2-1または2-2（つまり全加算器2）のクリティカルパスは、排他的論理和ゲート3が2段であるので、図7に示してあるWallace トリー型桁上げ保存加算器のクリティカルパスは、排他的論理和ゲート3が12段である。なお、図7において、全加算器2を示す枠の中に記載された数字は、部分積入力端子から当該全加算器2の出力端子までにおける排他的論理和ゲート3の合計通過段数を示すものである。以下の図においても、上記と同様に、部分積入力端子から当該全加算器2の出力端子までにおける排他的論理和ゲート3の合計通過段数を、全加算器2を示す枠の中に数字で記載してある。

【0008】 Wallace トリー型桁上げ保存加算器を用いて部分積を加算する場合、部分積の個数が19個までは、排他的論理和ゲート3の通過段数が12段分以内のクリティカルパスで加算できることが知られている。この回路については、たとえば、Kai Hwang 著、堀越ひさし監訳二階、「コンピュータの高速演算方式」、近代科学社、pp168-173、1980に記載されている。

【0009】 図10は、第2の従来例を示すブロック図である。

【0010】 この従来例において、部分積の個数が16

個以下である場合、全加算器2を2個組み合わせて4入力2出力加算器6を構成し、この4入力2出力加算器6を、Wallace トリ型桁上げ保存加算器の構成単位とすることができる。

【0011】第2の従来例は、4入力2出力加算器6同士の間の配線が規則的になるために、第1の従来例と比べて、レイアウトが容易となる利点がある。この回路については、たとえば、Mark R. Santoro, Mark A. Horowitz, "SPIM: A Pipelined 64 × 64-bit Iterative Multiplier", IEEE J. Solid-State Circuits, vol. SC-24, no. 2, pp487-493, Apr. 1989 に記載されている。

【0012】図11は、4入力2出力加算器の別の従来例を示す回路図である。

【0013】図11に示す4入力2出力加算器のクリティカルパスは、排他的論理和ゲート3の通過段数が3段であり、図10に示した4入力2出力加算器6のクリティカルパスに比べて、排他的論理和ゲート3の通過段数が1段少ない。したがって、図11の4入力2出力加算器をWallace トリ型桁上げ保存加算器の構成単位にすると、16個の部分積を加算する場合のクリティカルパス*20

$$\begin{aligned} X \cdot Y &= X \cdot (-2^{n-1} y_n + 2^{n-2} y_{n-1} + 2^{n-3} y_{n-2} + \dots + y_1) \\ &= X \cdot \sum_{i=0}^{n/2-1} (y_{2i} + y_{2i+1} - 2 y_{2i+2}) \cdot 2^{2i} \\ &= \sum_{i=0}^{n/2-1} PP_i \cdot 2^{2i} \end{aligned} \quad (1)$$

ここで、 $PP_i = (y_{2i} + y_{2i+1} - 2 y_{2i+2}) \cdot X$ は部分積である。この場合、(桁数/2) 個の部分積が生じる。

【0017】次に、Yの桁数が奇数である場合、符号桁 y_n を1ビット拡張し、 $Y = y_n y_{n-1} y_{n-2}$ とすることによって、桁数が偶数である場合と同様に扱うことができる。ただし、(桁数+1)/2 個の部分積が生※

$$\begin{aligned} X \cdot Y &= X \cdot (-2^{n'} y_{n'+1} + 2^{n'-1} y_n + 2^{n'-2} y_{n-1} + \dots + y_1) \\ &= X \cdot \sum_{i=-1}^{n'/2-1} (y_{2i+1} + y_{2i+2} - 2 y_{2i+3}) \cdot 2^{2i+1} \\ &= \sum_{i=-1}^{n'/2-1} PP_i \cdot 2^{2i+1} \end{aligned} \quad (2)$$

ここで、 $PP_i = (y_{2i+1} + y_{2i+2} - 2 y_{2i+3}) \cdot X$ は部分積である。ただし、 $y_0 = y_{-1} = 0$ であるため、最低次部分積 PP_{-1} は、 $PP_{-1} = (y_{-1} + y_0 - 2 y_1) \cdot X = -2 y_1 \cdot X$ となる。(1)、(2)式から、以下のことが分かる。

【0019】① PP_i は、Yの連続する3桁から得られる。

【0020】② PP_i の個数は、乗数Yの桁数が偶数のとき(桁数/2) 個、奇数のとき{(桁数+1)/2} 50

*スは、排他的論理和ゲート3の通過段数が9段になる。この回路については、たとえば、J. Mori, M. Nagamatsu, M. Hirano, S. Tanaka, M. Noda, Y. Toyoshima, K. Hashimoto, H. Hayashida, K. Maeguchi, "A10-ns 54×54-bit Parallel Structured Full Array Multiplier with 0.5-μm CMOS Technology", IEEE J. Solid-State Circuits, vol. SC-26, no. 4, pp600-606, Apr. 1991に記載されている。

【0014】ところで、2の補数表現のm桁の被乗数Xとn桁の乗数Yとを乗算する方法の1つに、2次のBooth のアルゴリズムを用いて部分積の個数を半分に削減し、配列型桁上げ保存加算器と2入力加算器とを用いて部分積を加算する方法が知られている。

【0015】つまり、まず、Yの桁数が偶数の場合、 $Y = y_n y_{n-1} y_{n-2} \dots y_1 y_0$ 、 $y_0 = 0$ ($y_n, y_{n-1}, \dots, y_1, y_0$ は、それぞれビットを示し 2^i づつ重み付がされている) とすると、2次のBooth のアルゴリズムは次式で表される。

【0016】

【数1】

※じる。また、Yの桁数が奇数である場合、 $Y = y_n y_{n-1} y_{n-2} \dots y_1 y_0$ 、 $y_0 = y_{-1} = 0$ とすることも可能であり、この場合、2次のBooth のアルゴリズムは次式で表される。

【0018】

【数2】

個である。

【0021】③ PP_i の値は0、±X、±2Xのいずれかである。 PP_i が-Xまたは-2Xとなった場合、Xの各桁のビットを反転し、LSBに1を加算する。

【0022】④最低次部分積から最高次部分積までの全部分積を 2^{2i} の重みづけをして加算する。

【0023】図12は、32桁の被乗数Xと32桁の乗数Yとを2次のBooth のアルゴリズムを用いて乗算する過程を示す説明図である。

5

【0024】Booth のアルゴリズムによってデコードした結果、部分積 PP_i が $-X$ または $-2X$ となった場合には負であること、すなわち符号桁が1であることを示すために、符号桁を上位桁に拡張する必要がある。そのため、図12における部分積 $PP_0 \sim PP_{14}$ の符号桁は拡張されている。また、部分積 PP_i が負になった場合にはLSBに1を加算する必要がある、図示したように $PN_0 \sim PN_{14}$ を加算する。したがって、31桁目は17ビットを加算することになる。なお、2次のBooth のアルゴリズムにては、たとえば、菅野卓雄監修、香山晋編、「超高速デジタルデバイスシリーズ、2、超高速MOSデナイス」、培風館、pp295-298、1986に記載されている。

【0025】

【発明が解決しようとする課題】上記従来例において、全加算器をWallace トリー型桁上げ保存加算器の構成単位とした場合に、 n 個の部分積を加算して和信号と桁上げ信号とを生成するときに、桁上げ保存加算器のクリティカルパスにおける排他的論理和ゲート3の通過段数は、 $\log_{(2/3)}(2/n)$ 段になり、すなわち、部分積が8個のとき8段、16個のとき12段、32個のとき16段になり、したがって、遅延時間が長くなるという問題がある。

【0026】一方、クリティカルパスにおける排他的論理和ゲート3の通過段数が3段である4入力2出力加算器をWallace トリー型桁上げ保存加算器の構成単位とした場合には、桁上げ保存加算器のクリティカルパスにおける排他的論理和ゲート3の通過段数は $\log_{(2/4)}(2/n)$ 段になり、すなわち、部分積が8個のとき6段、16個のとき9段、32個のとき12段になり、したがって遅延時間は短縮されるものの、ハードウェア量が増大するという問題がある。

【0027】さらに、乗数 Y の桁数が 2×2^k であるときに、乗数 Y を2次のBooth のアルゴリズムによってデコードすると、 2^k 個の部分積が生じ、この場合、部分積が負の場合にはLSBに1を加算する必要があるために、 $2^k + 1$ ビットを加算しなければならない桁が存在する。たとえば、32桁の被乗数 X と32桁の乗数 Y とを2次のBooth のアルゴリズムを用いて乗算する場合には、31桁目で17ビットを加算する必要がある、4入力2出力加算器を構成単位としてもクリティカルパスにおける排他的論理和ゲート3の通過段数の段数は10段になり、遅延時間が長くなるという問題がある。

【0028】本発明は、 n 個の部分積を加算して和信号と桁上げ信号とを生成するときに、全加算器をWallace トリー型桁上げ保存加算器の構成単位とした場合に比べてハードウェア量を増加させることがなく、4入力2出力加算器をWallace トリー型桁上げ保存加算器の構成単位とした場合と同様に規則的に配線することができ、さらにクリティカルパスにおける排他的論理和ゲートの通

6

過段数を $\log_{(2/4)}(2/n) + 1$ 段、すなわち、部分積が8個のとき7段、16個のとき10段、32個のとき13段と削減することができるWallace トリー型桁上げ保存加算器を提供することを目的とするものである。

【0029】また、本発明は、乗数 Y の桁数が 2×2^k であるときに、2次のBooth のアルゴリズムを用いて乗算を行った場合に、Wallace トリー型桁上げ保存加算器のクリティカルパスにおける排他的論理和ゲートの段数が増加しないこと、すなわち32桁の被乗数 X と32桁の乗数 Y とを乗算する場合に、クリティカルパスにおける排他的論理和ゲートの段数を10段にすることができるWallace トリー型桁上げ保存加算器を提供することを目的とするものである。

【0030】

【課題を解決するための手段】本発明は、 n ビットの入力信号を8ビットごとに分ける手段を設け、全加算器2個、または全加算器2個と半加算器1個とによって8入力6出力加算器を構成し、全加算器2個で上記8入力6出力加算器を構成する場合には、同一桁で加算される4ビットの信号と1桁上位で加算される2ビットの桁上げ信号とを出力し、一方、全加算器2個と半加算器1個とで上記8入力6出力加算器を構成する場合には、同一桁で加算される3ビットの信号と1桁上位で加算される3ビットの桁上げ信号とを出力する手段を設け、6入力3出力加算器が第1の全加算器と第2の全加算器と第3の全加算器とで構成され、上記6入力3出力加算器へ入力する6ビットの信号を、上記第1の全加算器と上記第2の全加算器とに送り、上記第1の全加算器の和信号を、上記第3の全加算器の3個の入力端子のうち遅延時間の長い2個の端子の一方に送り、上記第1の全加算器の桁上げ信号を、1桁上位のWallace トリー型桁上げ保存加算器内の上記6入力3出力加算器に送り、上記第3の全加算器の3個の入力端子のうち遅延時間の長い2個の端子の他方には、1桁下位のWallace トリー型桁上げ保存加算器内の6入力3出力加算器から送られてくる桁上げ信号を入力し、上記第2の全加算器の和信号または1桁下位のWallace トリー型桁上げ保存加算器内の6入力3出力加算器における第2の全加算器から送られてくる桁上げ信号を上記第3の全加算器における遅延時間の短い端子に入力する手段を設け、上記8入力6出力加算器を最上段に置き、この8入力6出力加算器の下に上記6入力3出力加算器を置き、この6入力3出力加算器2個から出力される6ビットの信号を下段の6入力3出力加算器に入力することを繰り返して上記入力信号を3ビットにまとめ、最後にその3ビットの信号を入力して和信号と桁上げ信号とを出力する第4の全加算器を設け、上記6入力3出力加算器に入力する6ビットの信号のうち、上段の上記8入力6出力加算器または6入力3出力加算器中の遅延時間の短い経路を通過する2ビットの信号を、上記6入力3出力加算器中の第1の全加算器の3個

の入力端子のうち遅延時間の長い2個の端子に入力する手段を設けたものである。

【0031】

【作用】本発明は、 n 個の部分積を加算して和信号と桁上げ信号とを生成するときに、全加算器をWallace トリー型桁上げ保存加算器の構成単位とした場合に比べてハードウェア量の増加がなく、クリティカルパスにおける排他的論理和ゲートの段数が $\log_{(2/4)}(2/n)+1$ 段、すなわち、部分積が8個のとき7段、16個のとき10段、32個のとき13段に削減することができ、また、4入力2出力加算器をWallace トリー型桁上げ保存加算器の構成単位とした場合と同様に、規則的に配線することができ、さらに、乗数 Y の桁数が 2×2^i であるときに、2次のBooth のアルゴリズムを用いて乗算を行った場合、Wallace トリー型桁上げ保存加算器のクリティカルパスにおける排他的論理和ゲートの段数が増加せず、たとえば、32桁の被乗数 X と32桁の乗数 Y とを乗算する場合にクリティカルパスにおける排他的論理和ゲートの段数は10段である。

【0032】

【実施例】図1は、本発明の第1の実施例である16個の部分積を加算するWallace トリー型桁上げ保存加算器を示すブロック図である。

【0033】図1に示す実施例は、 m 桁の被乗数と n 桁の乗数とを乗算する場合に用いられるWallace トリー型桁上げ保存加算器の例であり、 n ビットの入力信号を加算する過程で中間桁上げ信号が発生すると1桁上位のWallace トリー型桁上げ保存加算器に送り、1桁下位のWallace トリー型桁上げ保存加算器において中間桁上げ信号が発生するとそれも加算して最後に1ビットの和信号と1ビットの桁上げ信号を出力するWallace トリー型桁上げ保存加算器の例である。

【0034】また、この実施例では、16ビットの入力信号（つまり部分積 n ビット）が8ビットごとに分けられている。

【0035】8入力6出力加算器10は、2個の全加算器9、9で構成され、上記のように分けられた8ビットの信号を入力し、同一桁で加算される4ビットの信号と1桁上位で加算される2ビットの桁上げ信号とを出力するものである。

【0036】6入力3出力加算器11-1は、第1の全加算器9-1と、第2の全加算器9-2と、第3の全加算器9-3とで構成され、6入力3出力加算器11-1へ入力する6ビットの信号を、第1の全加算器9-1と第2の全加算器9-2とに送り、第1の全加算器9-1の和信号を、第3の全加算器9-3の3個の入力端子のうち遅延時間の長い2個の端子の一方の端子 b に送り、第1の全加算器9-1の桁上げ信号を、1桁上位のWallace トリー型桁上げ保存加算器内の6入力3出力加算器11に送り、第3の全加算器9-3の3個の入力端子の

うち遅延時間の長い2個の端子の他方の端子 a には、1桁下位のWallace トリー型桁上げ保存加算器内の6入力3出力加算器11から送られてくる桁上げ信号を入力し、第2の全加算器9-2の和信号を第3の全加算器9-3における遅延時間の短い端子 c に入力するものである。

【0037】6入力3出力加算器11-2、11-3は、6入力3出力加算器11-1と同様のものである。

【0038】また、上記実施例では、8入力6出力加算器10、10を最上段に置き、これら8入力6出力加算器10、10の下に、6入力3出力加算器11-1、11-2、11-3を置き、2つの6入力3出力加算器11-1、11-2から出力される6ビットの信号を下段の6入力3出力加算器11-3に入力することを繰り返して上記入力信号を3ビットにまとめている。

【0039】第4の全加算器9-4は、この3ビットの信号を入力して和信号と桁上げ信号とを出力する全加算器である。

【0040】6入力3出力加算器11（6入力3出力加算器11-1、11-2、11-3を総称したもの）に入力する6ビットの信号のうち、8入力6出力加算器10、10、または6入力3出力加算器11中の遅延時間の短い経路を通過する2ビットの信号（8ビットのうち、8入力6出力加算器10を通過するが、全加算器9を通過しない2ビットの信号、または第2の全加算器9-2の桁上げ信号 $C6$ と $C8$ との2ビットの信号）を、第1の全加算器9-1の3個の入力端子のうち遅延時間の長い2個の端子に入力している。

【0041】つまり、6入力3出力加算器11-1においては、8ビットのうち、8入力6出力加算器10を通過するが、全加算器9を通過しない2ビットの信号を、6入力3出力加算器11-1の第1の全加算器9-1の遅延時間の長い端子 a 、 b に入力し、6入力3出力加算器11-2においては、8ビットのうち、8入力6出力加算器10を通過するが、全加算器9を通過しない2ビットの信号を、6入力3出力加算器11-2の第1の全加算器9-1の遅延時間の長い端子 a 、 b に入力し、6入力3出力加算器11-3においては、第2の全加算器9-2の桁上げ信号 $C6$ と $C8$ との2ビットの信号を、1桁上の6入力3出力加算器11-3における第1の全加算器9-1の遅延時間の長い端子 a 、 b に入力している。

【0042】上記実施例において、全加算器9、9-1、9-2、9-3、9-4の回路そのものは、図7、図10の従来例で使用している全加算器2と同一の回路であり、つまり、図8に示す全加算器2-1または図9に示す全加算器2-2と同一の回路である。

【0043】ところで、図8に示す全加算器2-1の3個の入力端子のうち、入力端子 a または入力端子 b から入力して和を出力するパスにおける排他的論理和ゲート

3の通過段数は2段であるのに対して、入力端子cから入力して和を出力するパスにおける排他的論理和ゲート3の通過段数は1段である。図9に示す全加算器2-2についても、上記と同様であり、3個の入力端子のうち、入力端子aまたは入力端子bから入力して和を出力するパスにおける排他的論理和ゲート3の通過段数は2段であるのに対して、入力端子cから入力して和を出力するパスにおける排他的論理和ゲート3の通過段数は1段である。

【0044】図1における全加算器9、9-1、9-2、9-3、9-4も、上記と同様に、3個の入力端子のうち、入力端子aまたは入力端子bから入力して和を出力するパスにおける排他的論理和ゲート3の通過段数は2段であるのに対して、入力端子cから入力して和を出力するパスにおける排他的論理和ゲート3の通過段数は1段であり、入力端子cから入力して和を出力するパスにおける排他的論理和ゲート3の通過段数が少ないことに着目して、全加算器9-1、9-2、9-3、9-4を示す枠について、入力端子a、入力端子bよりも入力端子cを下げて描いてある。

【0045】また、上記実施例においては、入力端子cから入力して和信号または桁上げ信号を出力するパスの遅延時間が、入力端子aまたは入力端子bから入力して和信号または桁上げ信号を出力するパスの遅延時間の半分であることを利用することによって、クリティカルパスの遅延時間を削減するようにしている。

【0046】すなわち、まず、16個の部分積のビットを8ビットごとに2つのグループに分け、1つの8入力6出力加算器10を用いて、8ビットを6ビットの出力にまとめる。8入力6出力加算器10は、全加算器9を2個有し、2個の部分積は全加算器9を通らず、同一桁で加算される4ビットの信号（同一桁における6入力3出力加算器11-1または11-2に送られる信号）と、1桁上位で加算される2ビットの桁上げ信号（1桁上位における6入力3出力加算器11-1に送られる中間桁上げ出力信号C1、C2、または、1桁上位における6入力3出力加算器11-2に送られる中間桁上げ出力信号C3、C4）とを出力するものである。

【0047】次に、8入力6出力加算器10の出力6ビットを、6入力3出力加算器11-1、11-2によってそれぞれ3ビットにまとめる。6入力3出力加算器11-1は、第1の全加算器9-1、第2の全加算器9-2、第3の全加算器9-3を有し、6入力3出力加算器11-2も、第1の全加算器9-1、第2の全加算器9-2、第3の全加算器9-3を有し、6入力3出力加算器11-3も、第1の全加算器9-1、第2の全加算器9-2、第3の全加算器9-3を有する。

【0048】また、第1の全加算器9-1の和信号と桁上げ信号とが、第3の全加算器9-3の3個の入力端子のうち遅延時間の長い入力端子aと入力端子bとに接続

され、第2の全加算器9-2の和信号が、第3の全加算器9-3の遅延時間の短い端子cに接続されている。6入力3出力加算器11-1、11-2、11-3の出力信号は、第2の全加算器9-2の桁上げ信号、第3の全加算器9-3の和信号および桁上げ信号である。

【0049】ここで、6入力3出力加算器11-1に着目すると、6入力3出力加算器11-1中における第2の全加算器9-2の桁上げ信号C6が通過する排他的論理和ゲート3の段数は、他の2信号（桁上げ信号C9、6入力3出力加算器11-1から6入力3出力加算器11-3に向う信号）が通過する排他的論理和ゲート3の段数よりも1段少ない。上段の8入力6出力加算器10の出力6ビットのうち、全加算器9を通過しない2ビットを、6入力3出力加算器11中の第1の全加算器9-1における遅延時間の長い入力端子aと入力端子bとに送る。

【0050】続いて、2つの6入力3出力加算器11-1と11-2とにおける6ビットの出力信号を、下段の6入力3出力加算器11-3によって3ビットにまとめるが、6入力3出力加算器11-3中の第1の全加算器9-1における遅延時間の長い入力端子aと入力端子bとに、6入力3出力加算器11-1と11-2中における排他的論理和ゲート3の通過段数の少ない出力端子（すなわち第2の全加算器9-2の桁上げ信号を出力する端子）を接続する。

【0051】最後に、第4の全加算器9-4によって、上段に設けられている6入力3出力加算器11-3の出力3ビットを2ビットにまとめる。

【0052】16個の部分積を加算するWallace トリー型桁上げ保存加算器を、上記のように構成することによって、クリティカルパスにおける排他的論理和ゲート3の通過段数が10段になる。

【0053】このように、上記実施例によれば、全加算器をWallace トリー型桁上げ保存加算器の構成単位とした場合に比べてハードウェア量の増加がなく、クリティカルパスにおける排他的論理和ゲートの段数が $\log_{(2/3)}(2/n)+1$ 段、すなわち、部分積が8個のとき7段、16個のとき10段、32個のとき13段に削減することができる。一方、従来例においては、n個の部分積を加算して和信号と桁上げ信号を生成するときに、Wallace トリー型桁上げ保存加算器のクリティカルパスにおける排他的論理和ゲートの段数が $\log_{(2/3)}(2/n)$ 段、すなわち8個のとき8段、16個のとき12段、32個のとき16段であることと比較すると、上記実施例におけるクリティカルパスにおける排他的論理和ゲートの段数が少なく、遅延時間が短くなる。

【0054】また、上記実施例によれば、4入力2出力加算器を構成単位とした場合と同様に規則的に配線することができ、さらに、乗数Yの桁数が 2×2^k であるとき、2次のBooth のアルゴリズムを用いて乗算を行って

も、Wallace トリー型桁上げ保存加算器のクリティカルパスにおける排他的論理和ゲートの段数が増加せず、たとえば、32桁の被乗数Xと32桁の乗数Yとを乗算する場合にクリティカルパスにおける排他的論理和ゲートの段数は10段である。

【0055】上記実施例では、6入力3出力加算器11-1において、8ビットのうち、8入力6出力加算器10を通過するが、全加算器9を通過しない2ビットの信号を、6入力3出力加算器11-1の第1の全加算器9-1の遅延時間の長い端子a、bに入力し、6入力3出力加算器11-2において、8ビットのうち、8入力6出力加算器10を通過するが、全加算器9を通過しない2ビットの信号を、6入力3出力加算器11-2の第1の全加算器9-1の遅延時間の長い端子a、bに入力し、6入力3出力加算器11-3において、第2の全加算器9-2の桁上げ信号C6とC8との2ビットの信号を、1桁上の6入力3出力加算器11-3における第1の全加算器9-1の遅延時間の長い端子a、bに入力しているが、この代わりに、1桁下位のWallace トリー型桁上げ保存加算器内の6入力3出力加算器11-1における第2の全加算器9-2から送られてくる桁上げ信号C6を1桁上における第3の全加算器9-3における遅延時間の短い端子に入力するようにしてもよい。

【0056】つまり、6入力3出力加算器11-1において、8ビットのうち、8入力6出力加算器10を通過するが、全加算器9を通過しない2ビットの信号を、6入力3出力加算器11-1の第1の全加算器9-1の遅延時間の長い端子a、bに入力し、6入力3出力加算器11-2において、8ビットのうち、8入力6出力加算器10を通過するが、全加算器9を通過しない2ビットの信号を、6入力3出力加算器11-2の第1の全加算器9-1の遅延時間の長い端子a、bに入力し、6入力3出力加算器11-3において、第2の全加算器9-2の桁上げ信号C6の1ビットの信号を、1桁上の6入力3出力加算器11-1の第3の全加算器9-3の遅延時間の短い端子cに入力し、第2の全加算器9-2の桁上げ信号C8の1ビットの信号を、1桁上の6入力3出力加算器11-2の第3の全加算器9-3の遅延時間の短い端子cに入力し、6入力3出力加算器11-1の第2の全加算器9-2の和信号と、6入力3出力加算器11-2の第2の全加算器9-2の和信号とを、6入力3出力加算器11-3の第1の全加算器9-1の遅延時間の長い端子a、bに入力するようにしてもよい。

【0057】なお、上記実施例では、16ビットの入力信号を8ビットごとに分けているが、16ビット以外のnビットを8ビットごとに分けるようにしてもよい。

【0058】図2は、本発明の第2の実施例である32個の部分積を加算するWallace トリー型桁上げ保存加算器を示すブロック図である。

【0059】この実施例は、第1の実施例と同様に、6

個の6入力3出力加算器11をトリー型に組み合わせた構造を有するものである。

【0060】32個の部分積を加算するWallace トリー型桁上げ保存加算器を、図2のように構成することによって、クリティカルパスにおける排他的論理和ゲート3の通過段数が13段になる。

【0061】なお、図2においては、遅延時間の長い端子を示すa、bの符号と、遅延時間の短い端子を示すcの符号との表示を、図面を見やすくする都合上、省略してある。

【0062】図3～図6は、本発明の第3の実施例を示すブロック図であり、図3、図4、図5、図6は、32桁の被乗数Xと32桁の乗数Yとを2次のBooth のアルゴリズムを用いて乗算する場合に、それぞれ29桁目、30桁目、31桁目、32桁目のWallace トリー型桁上げ保存加算器を示すブロック図である。

【0063】図12で説明したと同様に、図3に示す29桁目において部分積 P_{Pi4} が負のときの補正項 P_{Ni4} を加算しなければならず、図5に示す31桁目において部分積 P_{Pi6} が負のときの補正項 P_{Ni6} を加算しなければならない。したがって、29桁目で16ビット、30桁目で15ビット、31桁目で17ビット、32桁目で16ビットを加算する必要がある。

【0064】なお、図3、図4から分かるように、29桁目と30桁目とは、図1に示したWallace トリー型桁上げ保存加算器の構造と同一である。図5、図6では、第1の8入力6出力加算器10以外に、第2の8入力6出力加算器13を用いて8ビットの入力を6ビットの出力にまとめる。8入力6出力加算器13は、1個の半加算器12と2個の全加算器9とを有し、3個の桁上げ信号のうち、1個の桁上げ信号に関する排他的論理和ゲート3の通過段数は1段である。また、1個の部分積は全加算器9を通らない。

【0065】図4から分かるように、30桁目の中間桁上げ出力信号はC1～C13であり、これらが31桁目の中間桁上げ入力信号D1～D13になる。したがって、図5から分かるように、6入力3出力加算器11-1において補正項 P_{Ni6} を加算することができる。この場合のクリティカルパスにおける排他的論理和ゲート3の通過段数は10段であり、図1に示した第1の実施例における16個の部分積を加算する場合と同一である。

【0066】なお、図5、図6において、8入力6出力加算器13は、2個の全加算器9、9と、1個の半加算器12とによって構成され、同一桁で加算される3ビットの信号（同一桁における6入力3出力加算器11-1に送られる信号）と、1桁上位で加算される3ビットの桁上げ信号（1桁上位における6入力3出力加算器11-1に送られる中間桁上げ出力信号C0、C1、C2）とを出力するものである。

【0067】また、上記実施例は乗算器において部分積

13

を加算する場合（ m 桁の被乗数と n 桁の乗数とを乗算する場合）のものであるが、これに限らず、複数の入力信号を一度に加算する複数のオペランド加算器（ m 桁の2進数を n 個加算する加算器、多入力加算器ともいう）においても、上記実施例Wallace トリー型桁上げ保存加算器を用いることができる。

【0068】

【発明の効果】本発明によれば、全加算器をWallace トリー型桁上げ保存加算器の構成単位とした場合に比べてハードウェア量の増加がなく、クリティカルパスにおける排他的論理和ゲートの段数が $\log_{(2/4)} (2/n)+1$ 段、つまり部分積が8個のとき7段、16個のとき10段、32個のとき13段に削減することができ、また、4入力2出力加算器をWallace トリー型桁上げ保存加算器の構成単位とした場合と同様に規則的に配線することができ、さらに、乗数 Y の桁数が 2×2^k であるとき、2次のBooth のアルゴリズムを用いて乗算を行っても、Wallace トリー型桁上げ保存加算器のクリティカルパスにおける排他的論理和ゲートの段数が増加せず、たとえば32桁の被乗数 X と32桁の乗数 Y とを乗算する場合にクリティカルパスにおける排他的論理和ゲートの段数は10段であるという効果を奏する。

【図面の簡単な説明】

【図1】本発明の第1の実施例である16個の部分積を加算するWallace トリー型桁上げ保存加算器を示すブロック図である。

【図2】本発明の第2の実施例である32個の部分積を加算するWallace トリー型桁上げ保存加算器を示すブロック図である。

【図3】本発明の第3の実施例を示すブロック図であり、32桁の被乗数 X と32桁の乗数 Y とを2次のBooth のアルゴリズムを用いて乗算する場合に、29桁目のWallace トリー型桁上げ保存加算器を示すブロック図である。

【図4】本発明の第3の実施例を示すブロック図であり、32桁の被乗数 X と32桁の乗数 Y とを2次のBooth のアルゴリズムを用いて乗算する場合に、30桁目のWallace トリー型桁上げ保存加算器を示すブロック図である。

14

【図5】本発明の第3の実施例を示すブロック図であり、32桁の被乗数 X と32桁の乗数 Y とを2次のBooth のアルゴリズムを用いて乗算する場合に、31桁目のWallace トリー型桁上げ保存加算器を示すブロック図である。

【図6】本発明の第3の実施例を示すブロック図であり、32桁の被乗数 X と32桁の乗数 Y とを2次のBooth のアルゴリズムを用いて乗算する場合に、32桁目のWallace トリー型桁上げ保存加算器を示すブロック図である。

【図7】第1の従来例であるWallace トリー型桁上げ保存加算器を示すブロック図である。

【図8】全加算器2の一例である全加算器2-1を示す回路図である。

【図9】全加算器2の他の例である全加算器2-2を示す回路図である。

【図10】第2の従来例を示すブロック図である。

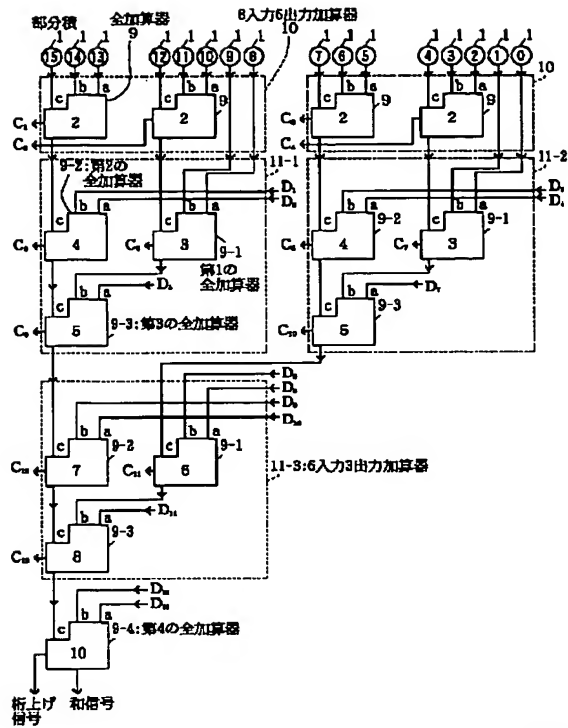
【図11】4入力2出力加算器の別の従来例を示す回路図である。

【図12】32桁の被乗数 X と32桁の乗数 Y とを2次のBooth のアルゴリズムを用いて乗算する過程を示す説明図である。

【符号の説明】

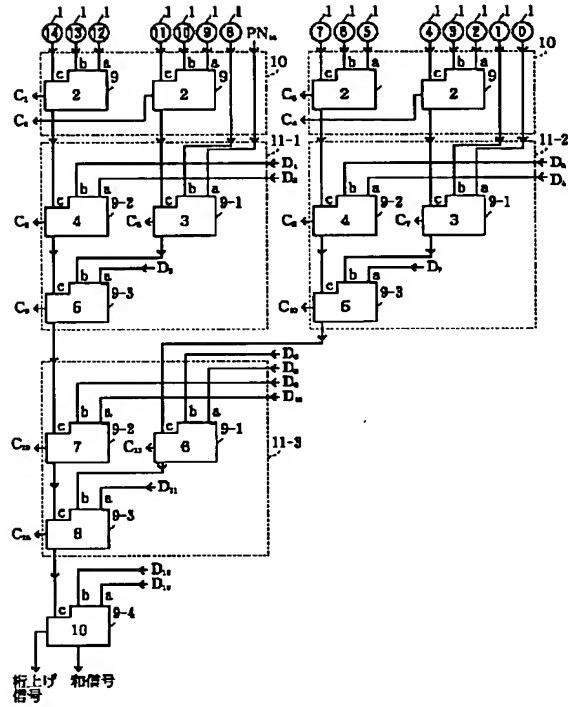
- 1…部分積、
- 2、2-1、2-2…全加算器、
- 3…排他的論理和ゲート
- 6…4入力2出力加算器、
- 9…全加算器、
- 9-1…第1の全加算器、
- 9-2…第2の全加算器、
- 9-3…第3の全加算器、
- 9-4…第4の全加算器、
- 10…8入力6出力加算器、
- 11、11-1、11-2、11-3…6入力3出力加算器、
- 12…半加算器、
- 13…8入力6出力加算器、
- C1～C13…1桁上位への中間桁上げ出力信号、
- D1～D13…1桁下位からの中間桁上げ入力信号。

【図1】



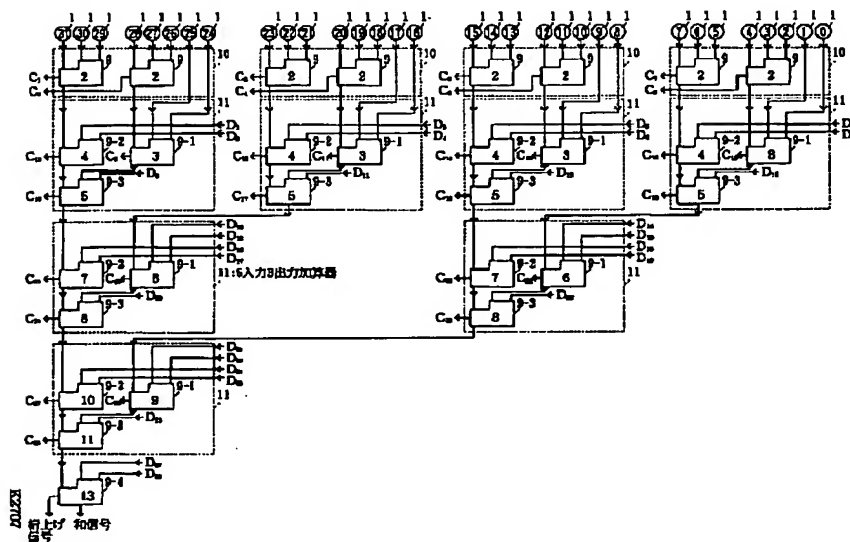
K2707

【図3】

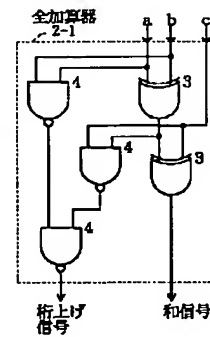


K2707

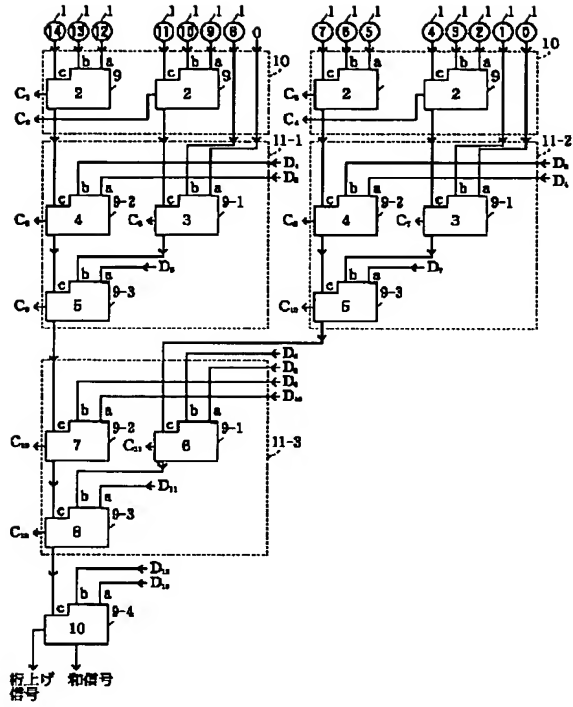
【図2】



【図8】

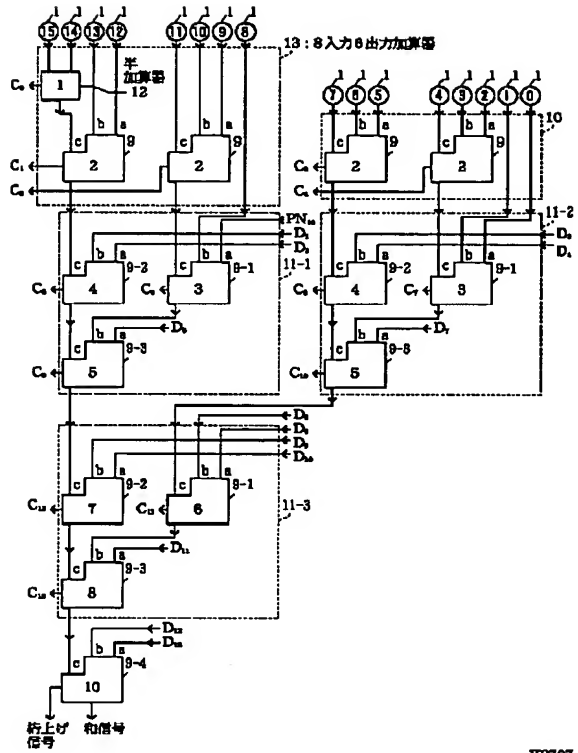


【図4】



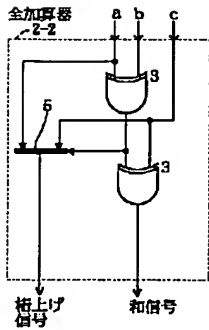
K2707

【図5】



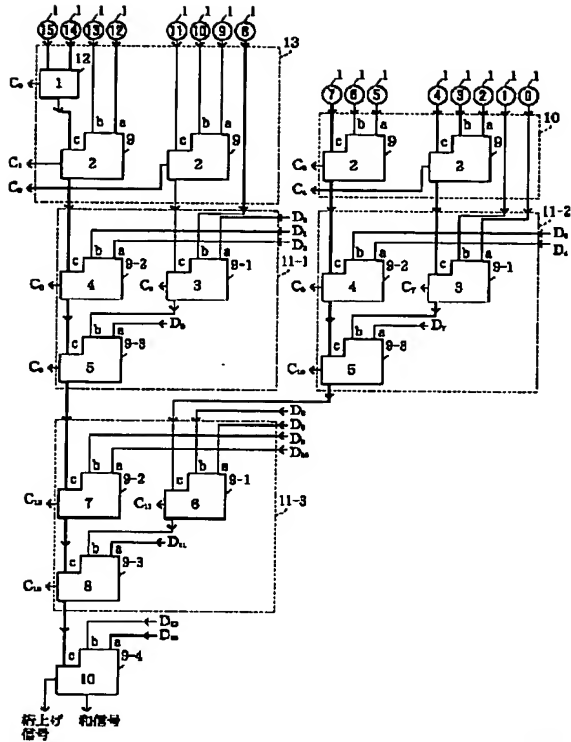
K2707

【図9】



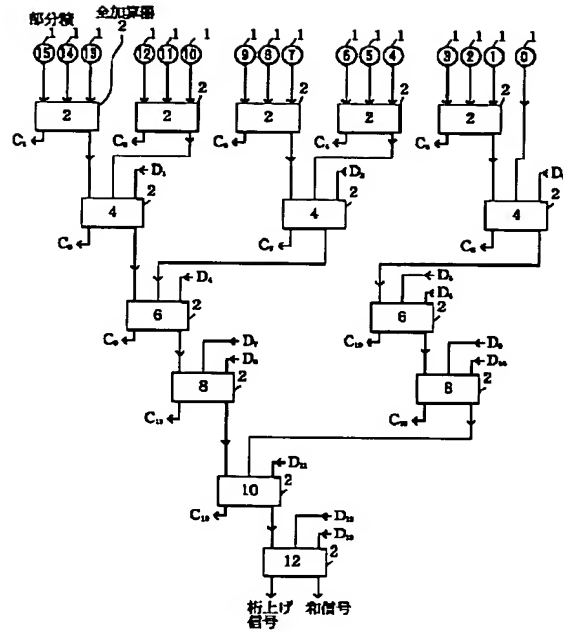
K2707

【図6】



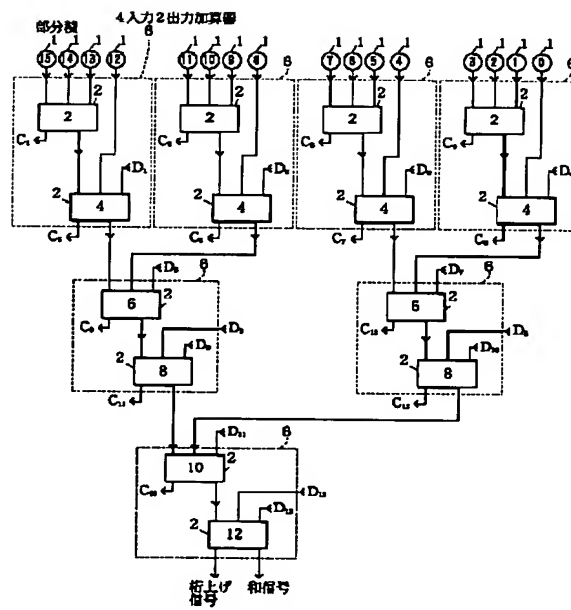
K2707

【図7】



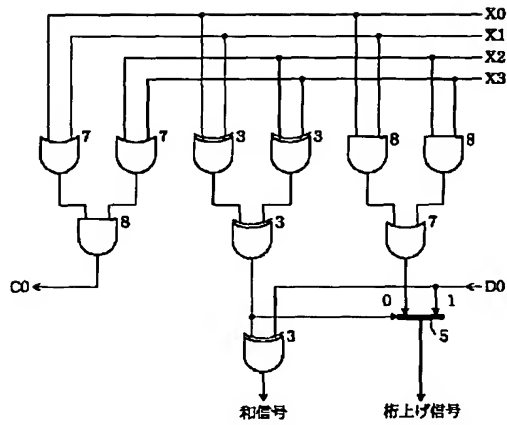
K2707

【図10】



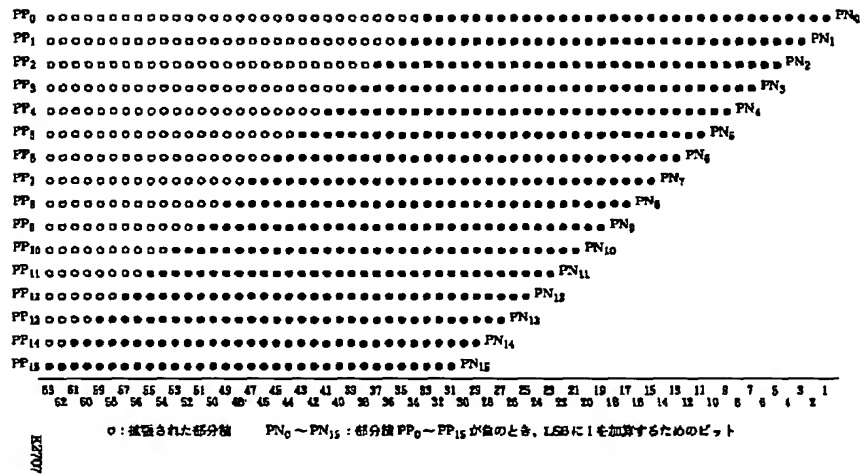
K2707

【図11】



K2707

【図12】



K2707